Google

Updating and Modernizing Clock Drivers

Chen-Yu Tsai <wenst@chromium.org>

蔡鎮宇 Tsai, Chen-Yu

- Kernel developer for platform bringup in ChromeOS
- Started as a hobbyist on Allwinner ARM-based SoCs in 2013
- Spoke about bring-up and maintainer experiences at OSS EU and ELCE

Disclaimer

Worked a lot on the Common Clock Framework.

But do not consider myself an expert.

Some Background

Clock API

- Common API for clock consumer drivers to follow
  - #include <linux/clk.h>
  - Implemented by each platform
- CLKDEV lookup code combined in 2010 & 2011

Early Days of Common Clock Framework

- Introduced in March 2011
- Initial implementations mostly had one device node per clock
  - Describe the full clock tree structure in the device tree
  - Bloated device trees and caused node address conflicts

Evolution of the Common Clock Framework

- Migration of existing platforms using the clock API to CCF
- Moved to one device node per clock controller
- Consumer (struct clk) and provider (struct clk_hw) separation
- Global clock lookup (CLKDEV) less common as platforms moved to DT
- More efficient clock parent lookup mechanisms implemented

What’s with Existing Drivers?

Drivers are upstreamed and forgotten

- Developers have moved on to the next platform
- Developers quit
- No real world users (in the community)

No Error Checking or Proper Cleanup

- Assumes clock driver will always succeed
  - Bad clock driver == non-working system
- Works poorly with retry under -EPROBE_DEFER
- Unloading module will likely blow up

Deprecated |struct clk| provider APIs

- clk_register*() family has ~1000 users
- clk_hw_register*() family has ~640 users
- Clearly still a lot of migration to be done
    - Documentation and guidance around this is poor

Inefficient String Matching for Clock Parent Matching

- String-based lookup against global clock tree
- Hash-based fast matching proposed [1]
  - Makes clk driver probe 30% faster
- Pointer and DT-based lookup added in 2019 by sboyd@kernel.org
  - Direct pointers for internal parents
  - DT-based clkspec (index into array) lookup

[1] https://lore.kernel.org/linux-mediatek/20211005065948.10092-1-mark-pk.tsai@mediatek.com/

What Should I Do?

Clean Up The Drivers

- Use DM instead of CLK_OF_DECLARE or init calls if possible
- Add proper error checking and error path cleanup
- Add .remove() driver callbacks

Migrate to clk_hw_register*

- Automated tools (sed, coccinelle) help
- Help the CCF maintainers out

Switch to Local Clock References for Parents

- Pointers for internal clocks
  - struct clk_hw *
- clkspec (device tree lookups) for external clocks
  - struct clk_parent_data
    - Use DT index instead of “clock-names” to avoid string matching
    - DT schema-based bindings force ordering of clocks
  - Can fall back to global clock names for DT compatibility

What We've Done

MediaTek Clock Driver Cleanup - Part 1

- Removal and cleanup code added for driver library
  - One SoC (MT8195) driver cleaned up
  - Future drivers follow example set
- TODO - clean up all the other SoC drivers

https://lore.kernel.org/linux-clk/20220122091731.283592-1-wenst@chromium.org/

Migrate MediaTek to clk_hw_register*

- Massive patch series rewriting existing code
- Split into multiple patches doing either manual or automated rewrite
  - Easier review
  - Avoid hitting email size limit on linux-clk mailing list
- Squashed together after review for bisectability

https://lore.kernel.org/linux-clk/20220510104804.544597-1-wenst@chromium.org/

Switch to Local Parent References Where Possible

- Code working; haven’t been cleaned up nor posted
- Probe and boot time improvement
  - Not much at the moment
  - MT8192 has ~600 clocks
  - Checking for duplicate clock names actually takes up a lot of time

CCF Future Work & Wish List

Improve Duplicate Clock Check

- Hash table
- Let clock driver opt out of string based check
- Drop clock names altogether?
  - Still useful for debugging though

Improve Documentation

- Make the clk core more approachable
  - Better kerneldocs
- An introduction doc maybe?

Kunit Tests

Extend test coverage of the CCF with more use cases.

.get_parent Rework

- Returns u8
  - No way to signal error
  - Limited to 256 parents
- Have .get_parent op return a clk_hw*

Merge Multiple Regmap Clock Implementations

- Multiple variants of regmap-based basic clock types
  - qcom, mediatek, …
- Merge and promote the basic stuff to the core

Prepare Lock Problems

- One BIG LOCK
  - Blocks unrelated clocks from changes
  - Split up the lock or make it smaller
    - Lock just the affected subtree?
- Lockdep complains about deadlocks when used on slow busses

Questions?